1. (10 points) Consider a switching system which uses a crossbar with virtual output queues and a LOOFA scheduler. Suppose that this system implements multicast by having the input ports copy arriving cells into virtual output queues when they arrive. Assume that the maximum fanout of a multicast cell is 3. If \( c \) is a cell in a VOQ at the start of an arrival phase, what is the maximum amount by which \( \text{slack}(c) \) can change during the arrival phase? Explain.

During an arrival phase, a multicast cell can arrive and be copied to three VOQs. If all three copies of the cell precede cell \( c \) in the ordering of cells at the input, then \( p(c) \) can increase by 3, which means that \( \text{slack}(c) \) will decrease by 3.

How large must the speedup be to ensure that the system is work-conserving? Explain.

If there is a speedup of 4, then we can have four transfer phases for every arrival and departure phase. This is enough to ensure that \( \text{slack}(c) \) cannot decrease as time progresses, since each transfer phase increases \( \text{slack}(c) \) by at least 1, while each arrival phase decreases \( \text{slack} \) by at most 3 and each departure phase decreases \( \text{slack} \) by at most 1.

2. (15 points) The CCF algorithm for crossbar scheduling uses a stable matching, instead of a maximal matching. The algorithm for finding a stable matching described in the notes is reproduced below. Explain why the algorithm does produce a stable matching.

Given, sets \( A \) and \( B \), where each element of \( A \) assigns a rank to each element of \( B \) and each element of \( B \) assigns a rank to each element of \( A \).

- Initially all elements are unmatched
- Repeat until all elements have been matched
  - Each unpaired element in \( A \) makes a “bid” for the top-ranked member of \( B \) for which it has not bid previously.
  - Each element of \( B \) selects the best of its current bids or stays with its current “partner” (if any).
  - The selections made by the elements of \( B \) form the current matching.

Assume, to the contrary, that the constructed matching is not stable. This means that there are two pairs \((a_1, b_1)\) and \((a_2, b_2)\) such that \( a_1 \) prefers \( b_2 \) to \( b_1 \) and \( b_2 \) prefers \( a_1 \) to \( a_2 \). If this were the case, then \( a_1 \) must have made a bid for \( b_2 \) before it bid for \( b_1 \). Since the members of \( B \) only change partners to improve their preference, at the time that \( a_1 \) bid for \( b_2 \), either \( b_2 \) was unmatched or it was matched with a partner that it ranked no higher than \( a_2 \). In this case, \( b_2 \) must have switched partners, matching it with \( a_1 \). But this yields a contradiction, since \( b_2 \) would not have switched from \( a_1 \) to \( a_2 \), after becoming matched with \( a_1 \).
3. (20 points) Consider a multistage network in which each switch element has eight inputs and eight outputs, and there is a shared buffer that can hold 400 cells. Suppose that each output has a reserved area of 20 cells that only it can use. The remaining buffer space forms a shared pool than can be used by any output. Let $q_i$ is the number of cells currently queued for output $i$. Give an expression for the amount of unused space in the reserved area for output $i$.

$$\max(0, 20 - q_i)$$

Give an expression for the amount of space that output $i$ is using from the shared buffer pool.

$$\max(0, q_i - 20)$$

Assume that the system uses simple grant flow control. That is, if an upstream neighbor gets a grant, it can send any cell that it has for the downstream neighbor. What condition must the $q_i$ satisfy in order to make it safe to issue upstream grants to all upstream neighbors?

For all upstream inputs to get grants, each output must be able to receive eight cells. The worst-case is where all the upstream cells go to a single output, so for each output, the number of available slots in its reserved space plus the number of available slots in the shared pool must be at least 8. So, we need to satisfy the following inequality for all $i$.

$$\left(240 - \sum_j \max(0, q_j - 20)\right) + \max(0, 20 - q_i) \geq 8$$

Explain how with simple grant flow control, a single greedy output can block traffic from reaching other outputs. How can this problem be corrected?

A greedy output can fill its reserved space and the shared buffer. This can prevent the switch element from issuing grants to the upstream neighbors, blocking traffic even to those outputs that have space available in their reserved areas. To correct this, we need to allow cells for the uncongested outputs to be forwarded from the upstream neighbor, while cells for the congested output are held back. This requires that switch elements issue separate grants for each of their outputs. An alternative approach is to use acknowledgement flow control, instead of grant flow control. This is simpler, but does not work quite as well.

4. (9 points) Fill in the following table.

<table>
<thead>
<tr>
<th>$x$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_{2,5}(x)$</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>3</td>
<td>8</td>
<td>4</td>
<td>9</td>
</tr>
</tbody>
</table>

What is $\sigma(2, 73)$ in $D_{125, 5}$? $73 = 243_5$ so $\sigma(2, 125) = 234_5 = 69$.

What is $\sigma(2, 89)$ in $Y_{243, 7}$? $89 = 10022_3$ so $\sigma(2, 89) = 12020_8 = 141$.
5. (10 points) Explain why the two networks shown below are strongly isomorphic. Hint: recall that $R_{k,d}D_{n,d}$ is strongly isomorphic to $Y_{n,d}$, where $R_{k,d}$ is the digit reversal permutation.

In the top network, two subnetworks have been outlined and identified. The first is $R_{3,2}$ and the second is $D'_{8,2}$ which is strongly isomorphic to $D_{8,2}$ meaning that the combination of these two is strongly isomorphic to $Y_{8,2}$. In the bottom network, a subnetwork $Y_{8,2}$ has been outlined. Since a strongly isomorphic subnetwork can be substituted in another without changing the overall structure of any network containing it, the two networks are isomorphic.

6. (6 points). Consider the network $D_{16,2} \otimes Y_{27,3} \otimes D_{16,4}$ and let $L$ be some link in stage 5. What is the maximum load that any input/output session can place on $L$, assuming that the maximum load on any input or output is 1?

1/16

How many different inputs can send traffic that passes through $L$?

48

What is the maximum load that can be placed on link $L$ by any set of input/output sessions that don't overload the inputs or outputs of the network?

3
7. (10 points) We have studied two types of fixed threshold resequencers, referred to as strict and loose. Explain the differences between the two and briefly discuss the relative advantages and disadvantages of each.

In a strict resequencer, arriving cells that are “older” than the age threshold are discarded. In a loose resequencer, these cells are added to the resequencing buffer and forwarded in the order of their timestamps.

The strict resequencer has two advantages. First, it never forwards cells in the incorrect order. Second, it is very simple to implement, using a simple timing wheel data structure. The loose resequencer provides better performance for bursty traffic. While it can forward cells out of order, it is often able to correctly resequence and forward cells that would be discarded by a strict resequencer.

The disadvantage of a loose resequencer is that it is somewhat more complicated to implement, requiring a larger timing wheel, a separate “lag pointer” and a mechanism for quickly advancing the lag pointer past empty buffer slots.

8. (8 points) The figure below shows a network flow problem corresponding to a distributed scheduling problem. Fill in the blanks to specify a blocking flow corresponding to a BLOOFA schedule. Assume that output 0 has the smallest backlog, then outputs 1, 2 and 3.